

テスト設計コンテスト2018 決勝

テスト設計成果物の紹介

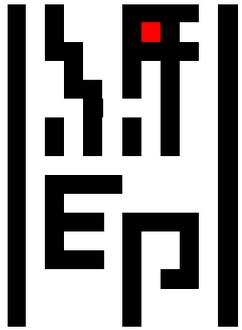
チーム

「紙印テスト倶楽部」

メンバー

小田部

チーム紹介



チーム名 **「紙印テスト倶楽部」**

コミケと技術書展にて**テスト設計本**を
頒布

新たなネタを披露しに来ました



メンバー紹介

小田部

テスコン6年目にして

安定のボッチ

テスト設計の目標

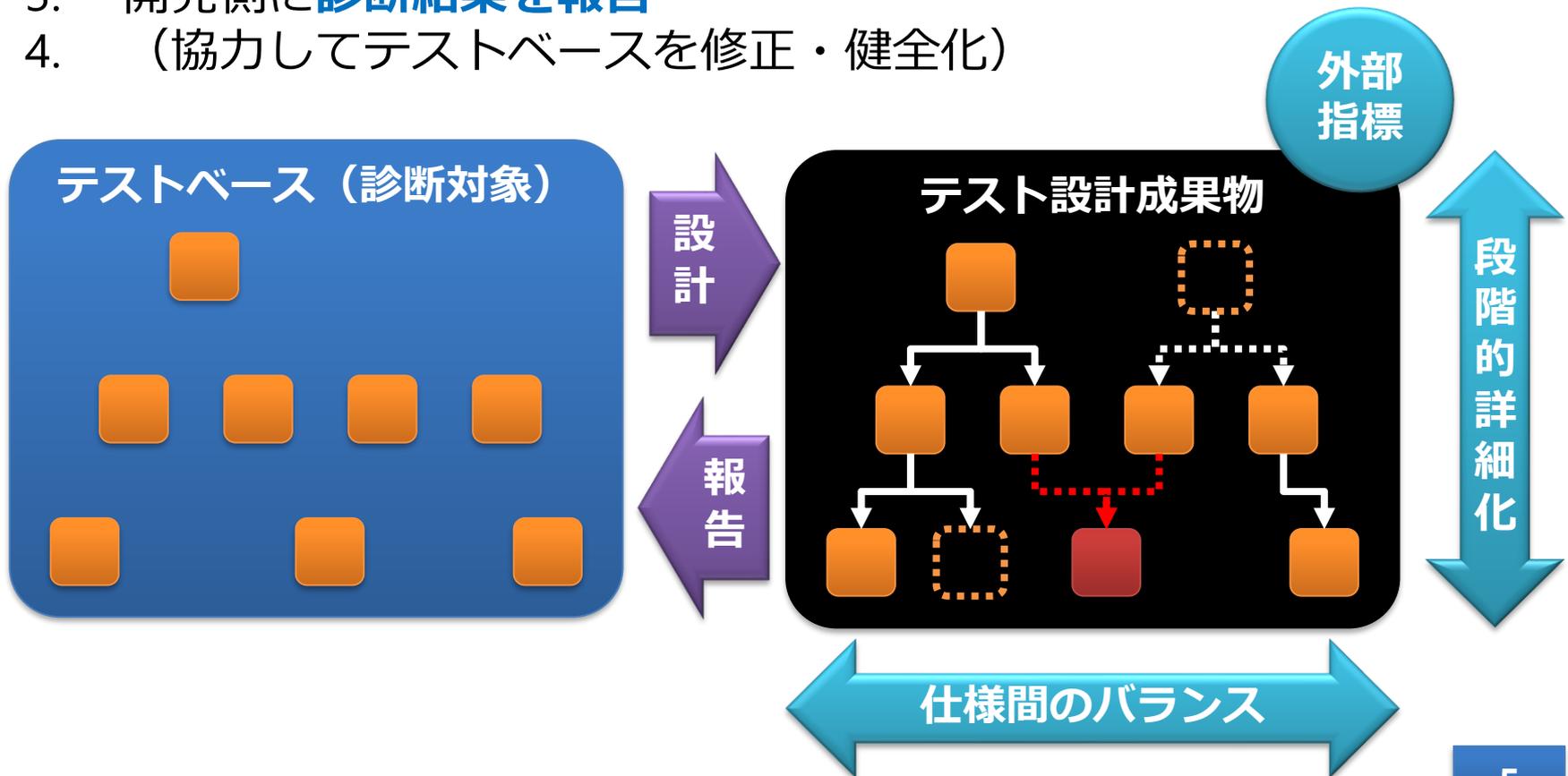
- **「テストベースの健全化」**が目標
 - 問題のあるテストベースは人を不幸にする
 - 開発されるのは**「問題のあるシステム」**
 - テスト開発で得られるのは**「問題のあるテストケース」**
 - テストベースの診断にはテスト設計成果物（≠テストケース）が最適である
 - テストベースとその関連情報を様々な観点でテスト設計成果物に変換する過程で、テストベースの問題点が浮き彫りになる
 - **テスト実施前に品質を確保**するのが紙印テスト倶楽部流のテスト設計
 - テストベースで定義された仕様空間をテスト設計成果物で診断し、開発されたシステムのテスト空間をテストケースを用いて網羅する

診断テスト (Diagnostic Test)

- 以下の方針でテストベースが健全か診断する
 - **仕様空間の網羅**
 - テスト設計を通じて仕様空間（テストベースで定義すべき仕様の集合）に不足や問題がないか網羅的に診断する
 - **診断方針の設定**
 - 仕様空間に対して適切な診断方針を設定する
 - **プロジェクトの常時診断**
 - 開発の進捗に合わせてリアルタイムでテストベースの診断結果を開発側にフィードバックする
- 上記のテスト手法を「**診断テスト**」と命名
 - 仕様空間を適切に診断し、その結果を速やかに開発側にフィードバックすることでテストベースの健全化をサポートする

診断テスト概要

1. テストベースを**テスト設計成果物**に変換
2. 垂直、水平、指標の**各診断の実施**
3. 開発側に**診断結果を報告**
4. (協力してテストベースを修正・健全化)



診断結果

• 垂直診断

- 自社の利益獲得にはカラオケ店オーナーによるカラオケシステムの購入が必須条件だが、現状のテストベース通りのカラオケシステムが完成しても**オーナーの購入動機に結び付かない**
- 高音質・高画質なカラオケシステムは他社でも実現が容易なので、**製品の強みとなりにくい**
- フラッグシップモデルの成功に欠かせない魅力的な**有料コンテンツの内容が不明**
- ナイト店のユーザはカラオケ利用が目的でない場合も十分あり得るので、その状況から**カラオケ利用への導線を確立**することが重要課題

• 水平診断

- **古い外部機器（特にリモコン）との互換性**を確保すると、快適でリッチな体験から遠ざかる
- 大容量通信の使用を前提としたフラッグシップモデルは**NBのナイト店に向いてない**

• 指標診断

- ユーザビリティを無視した、**全台再起動が必須**のオーナー設定
- 運用性に影響する、**大容量通信のインフラ整備**とその維持

テスト設計サイクル

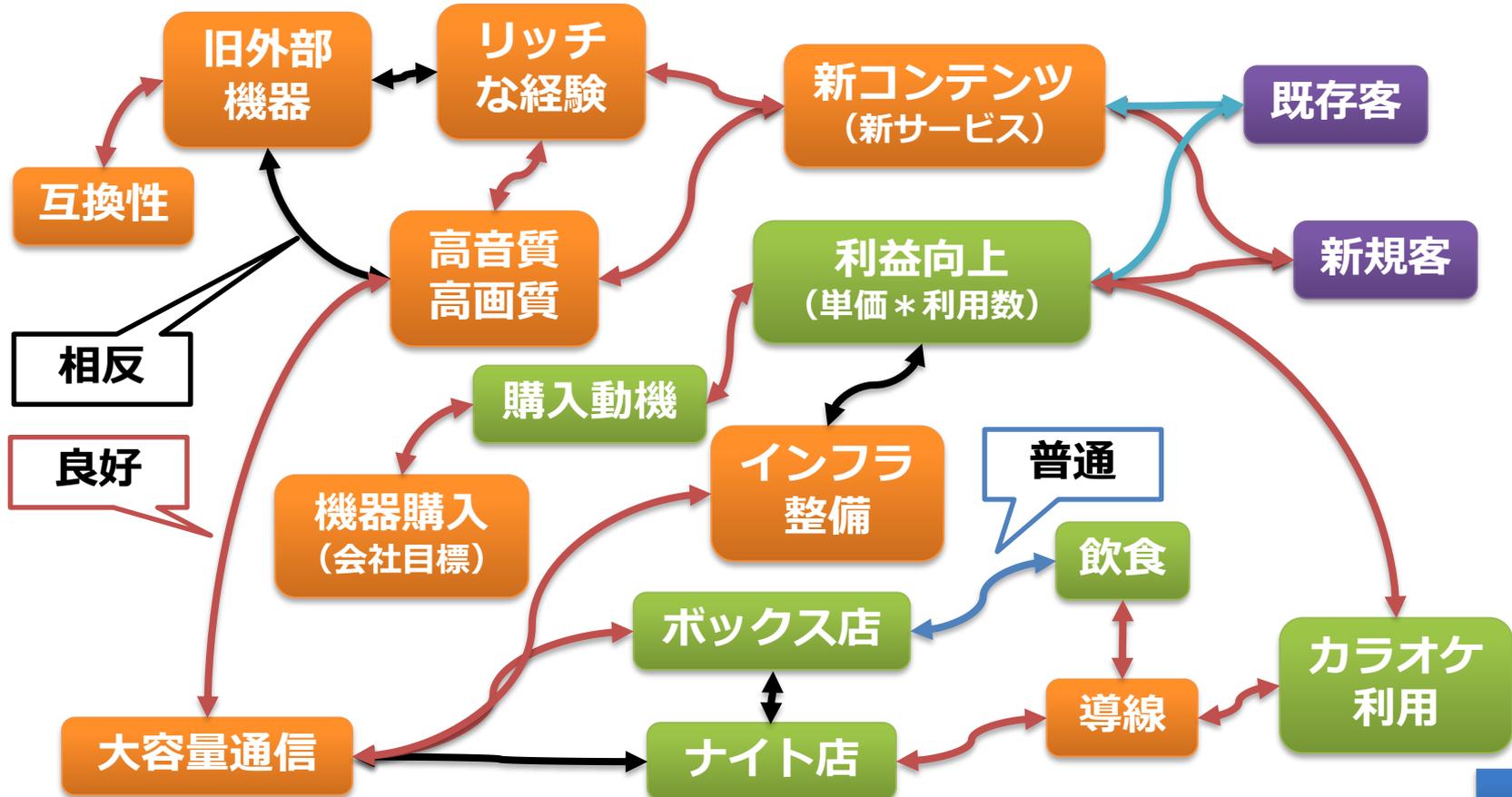
- テスト設計を**2種類のサイクル**で実施することで常時診断を実現
 - **職人型**のテスト設計サイクル
 - 仕様を分析し、**テスト観点および因子と水準を抽出**する
 - 高速かつ高精度なテスト設計が重要
 - 得られた知見の標準化や共有が重要
 - **探検型**のテスト設計サイクル
 - 「無いもの」や「矛盾」など、**「リスク」を可視化**する
 - これまでの診断結果から、次に実施するテスト設計の内容を決定する
 - 例：全体把握 → 資料の有無 → 記述の有無 → 記述粒度
 - 適切な**テスト設計内容の設定**が重要

要求同士の相関調査

機器購入実現の流れを想定

- 新コンテンツがカギだが現状不明
- 新規客の獲得が重要

- ボックス店とナイト店で要求が異なる
- 旧機器との互換性確保は逆効果
- ナイト店は導線確保が課題



Fujiメソッド概要



ステークホルダ分析

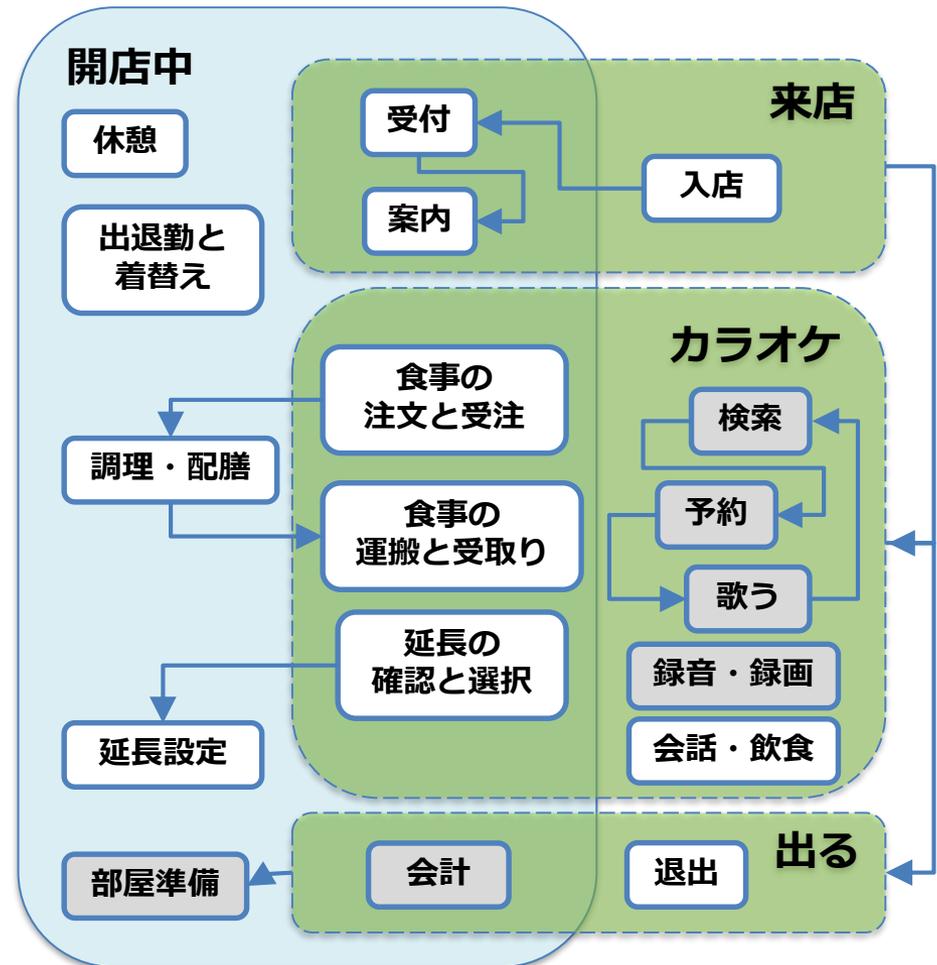


		当り前	魅力	反対
ボックス店	個人	曲の調整 練習環境	歌唱評価 全国比較	誤評価 曲が少ない
	団体	簡単操作 見やすい画面	曲が充実 演出盛上がり	使いにくい 曲が少ない
	オーナー	一括設定 手間いらず	人気有料コンテンツ 専用の曲と映像	飽きられる
	新規	高音質 高画質	会員専用コンテンツ ライブ並みの臨場感	予約取れない 少ない専用コンテンツ
ナイト店	個人	課金明確 曲がある	周囲の注目 名音声フィルタ	課金トラブル 他の客が占有
	オーナー	一括設定 手間いらず	人気有料コンテンツ オンラインで快適操作	課金トラブル 客同士トラブル
サプライヤ		機器の設定 しやすさ	自動接続 遠隔監視	不明な設定トラブル 解決しにくい

ユーザーアクションの分析



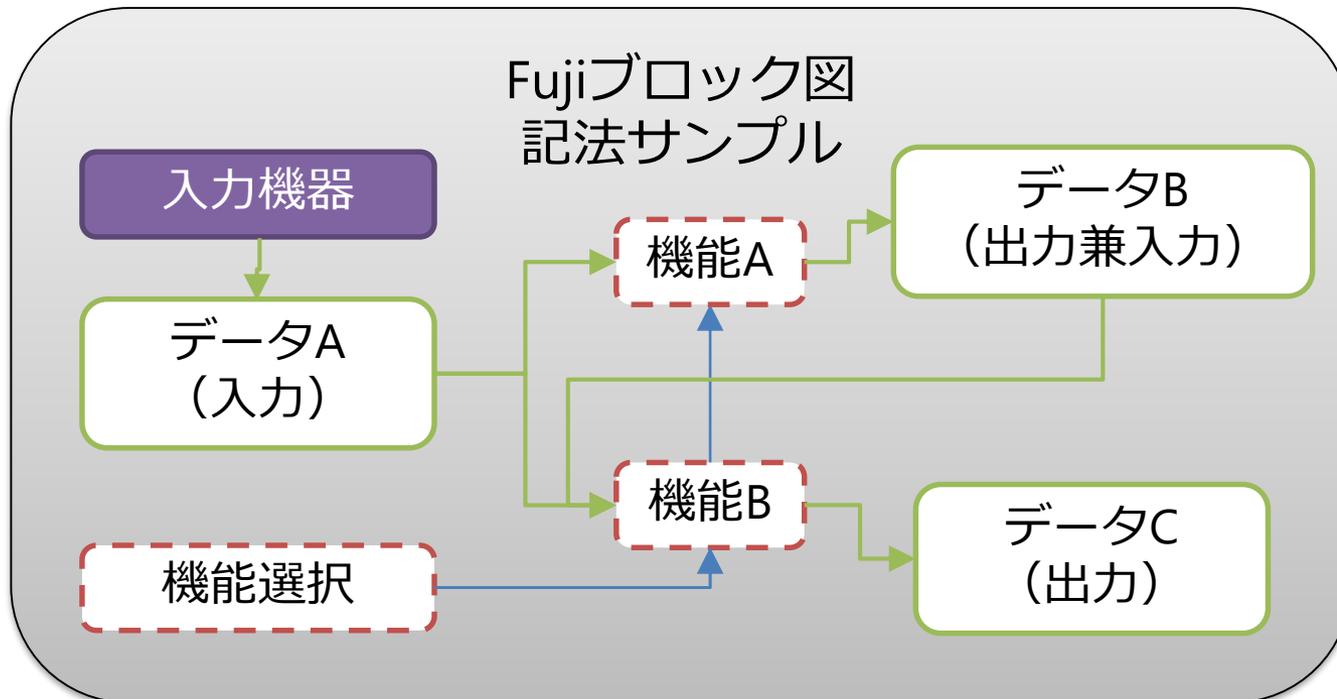
- テスト対象の利用を含む、**各ステークホルダの行動を想定**する
 - 開店中のカラオケ店を想定
- ステークホルダの行動にテスト対象の**システムが対応できるか**分析する
 - 「部屋準備」を早く終わらせるには、**一括標準設定機能**が欲しい
 - **録画中に店員の割込み**が入ったら嫌だろうな
 - 使い方を間違えたときに復帰出来るか？



モデル化 (Fujiブロック図)



- テストベースを**個々の機能とその入出力でモデル化**
 - 個々の機能単位での**軽量・高速・柔軟なテスト設計**を可能にした



発散分析



分類整理を前提としたマインドマップ

- ラルフチャートと論理的機能構造を参考に、メインブランチを設定



集約分析と分類整理



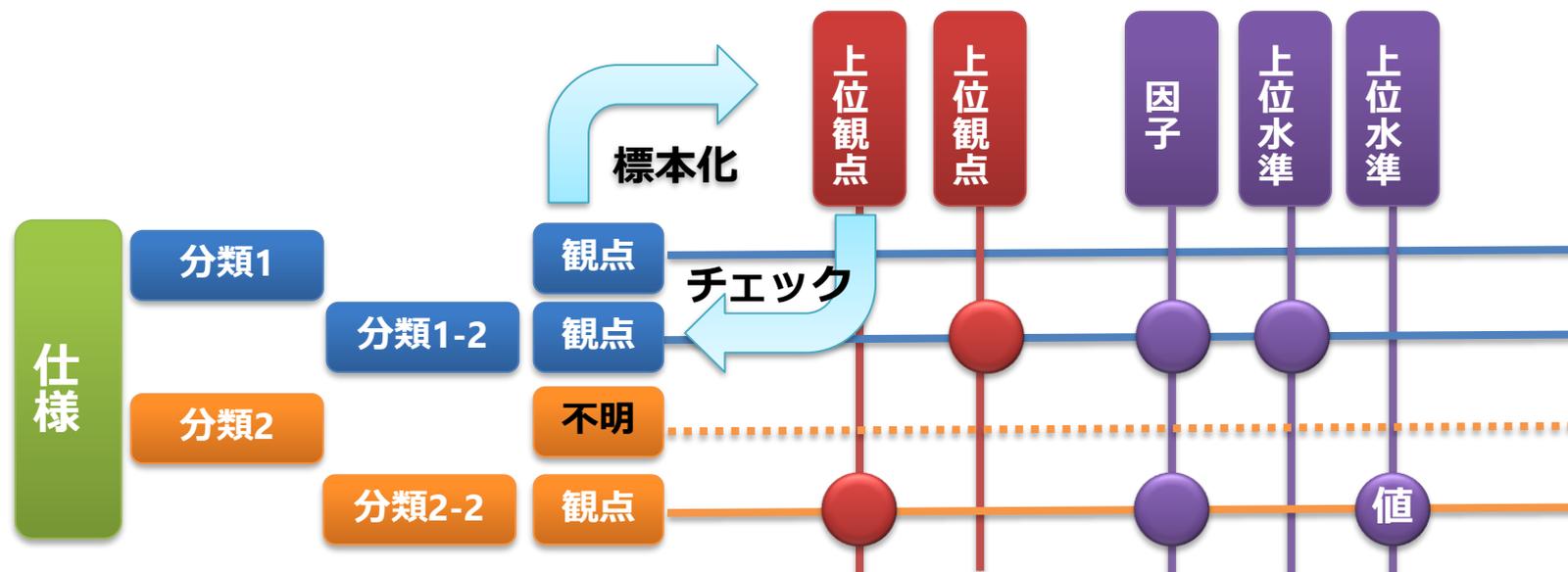
ツリー構造で分類整理

- 抜け漏れのチェックと再分析および「操作」「テストタイプ」「テストレベル」のタグ情報を追加し、テスト実装の内容を詳細化



観点および水準の標本化

- 各テスト設計サイクルでは以下の手順で**分析漏れを軽減**し**設計精度を向上および平準化**する
 - 分析結果から共通の観点および水準を**標本化**する
 - 分析時は**上位観点**および**上位水準**も用いて分析する
 - 仕様が曖昧なら「●」、詳細なら「値（水準）」を入れる
- 仕様、質問、観点、因子、水準をまとめて**設計効率を向上



フィルタリングと集約



全分析結果をフィルタリングで集約

- 全分析結果を「操作・作業」「テストタイプ」「テストレベル」でフィルタリングし、集約された情報を用いてテスト実装

Func

Type

Lv

整理結果A

整理結果B

整理結果C

整理結果D

整理結果E

フィルタリング・集約

- ◆ **テスト実装に必要な情報のみを抽出**
- ◆ **最も知見が蓄積された時点の情報**を利用できる
- ◆ **重複した情報**もまとめてテスト実装

逆さFujiメソッド

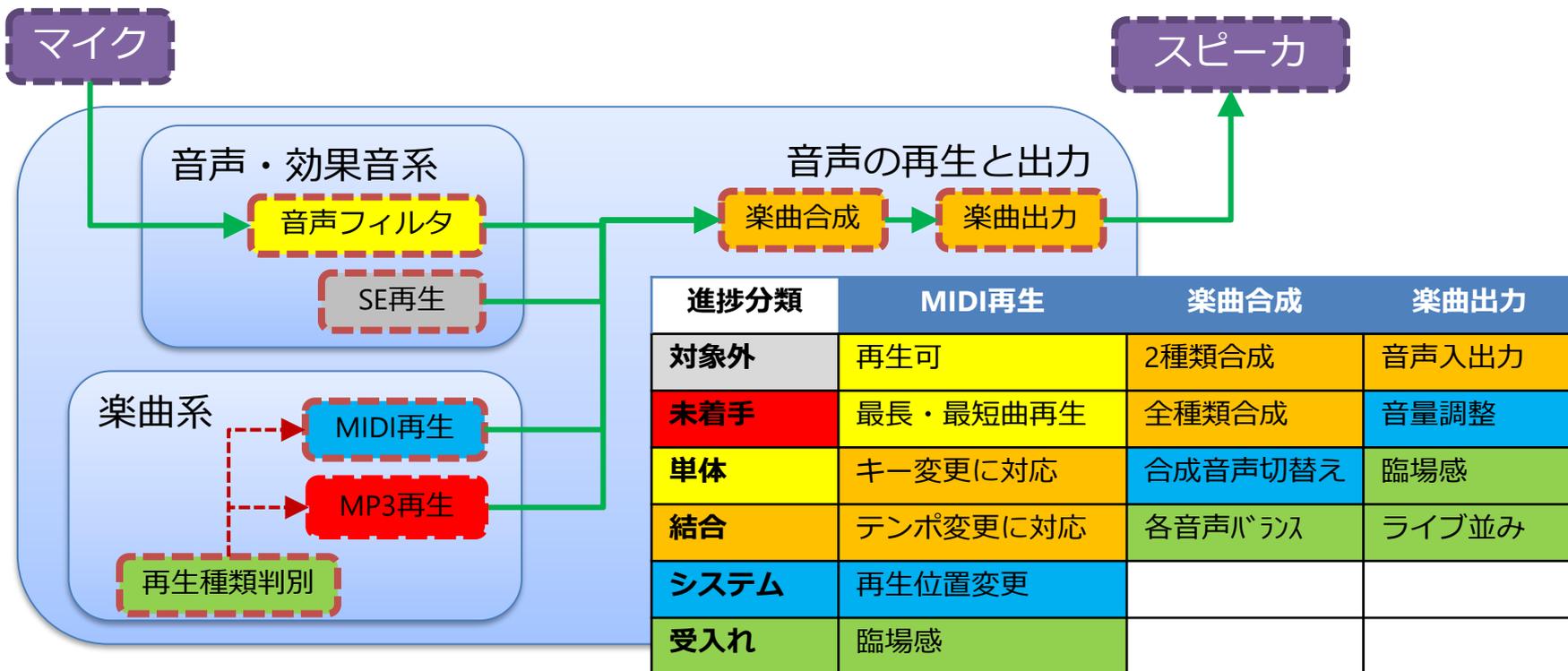


- 引き上げ操作**不具合の想定原因**
 - ネットワーク**エラー検知**機能の動作不良
 - 他の操作が**並行して実行**可能だった
 - **データセンター**側の不具合
 - テストーちゃん (サプライヤ) の**操作ミス & 報告漏れ**
 - **手順書**の記述漏れや分かりにくさ等の不備
 - 引き上げ操作の**二重実行**が可能だった
 - 引き上げ操作中に**自動処理**が実行される
 - 引き上げ操作エラー時の**リカバリ**手段がない
 - カラオケシステムの**故障**
 - **ネットワーク設定**ミス



テストアーキテクチャ設計

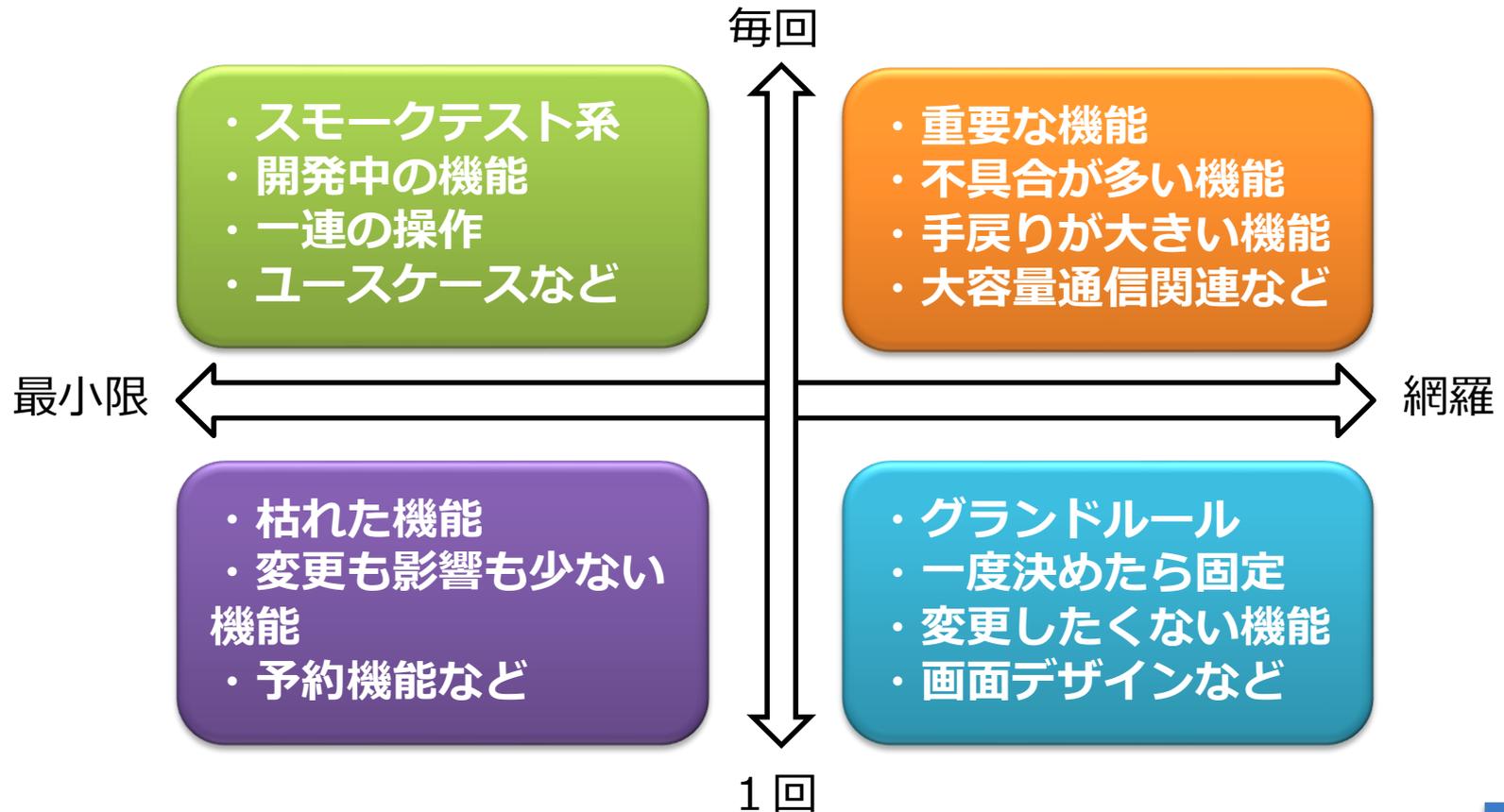
- 開発に合わせたテストアーキテクチャ
 - 開発とテストの進捗から次に実施するテストを決定
 - 各機能の品質レベルも確認



テスト実施の品質

- 実施内容を **4カテゴリで分類**

- 不具合の発生状況に応じて所属するカテゴリを変更する



テスト設計まとめ

- **テストベースの健全化**
 - テスト実施前に品質を確保
 - 診断テストの実施
- **テスト設計サイクル**
 - 短いサイクルでテスト設計を繰り返し実施する
 - 知見の蓄積、標準化、水平展開
- **2種類のテスト設計手法と設計支援ツール**
 - 使い分けることによるテスト設計の完成度向上
 - テスターちゃん大人気♡

参考文献

- 唯野 奈津実著：「カラオケ上達100の裏ワザ」
- 湯本 剛氏 発表資料：「テストアーキテクチャの具体例」
(<http://jasst.jp/symposium/jasst12tokyo/pdf/A2-6.pdf>)
- 小田部 健著：「テスト設計入門 2017年度版」

